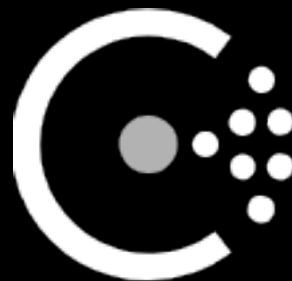
20명 규모의 팀에서 Vault 사용하기

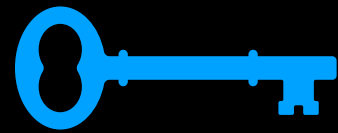
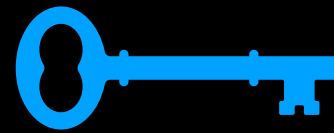
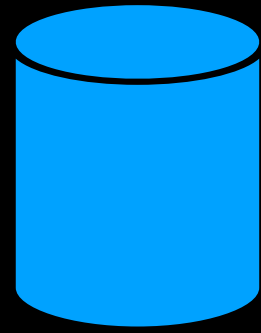
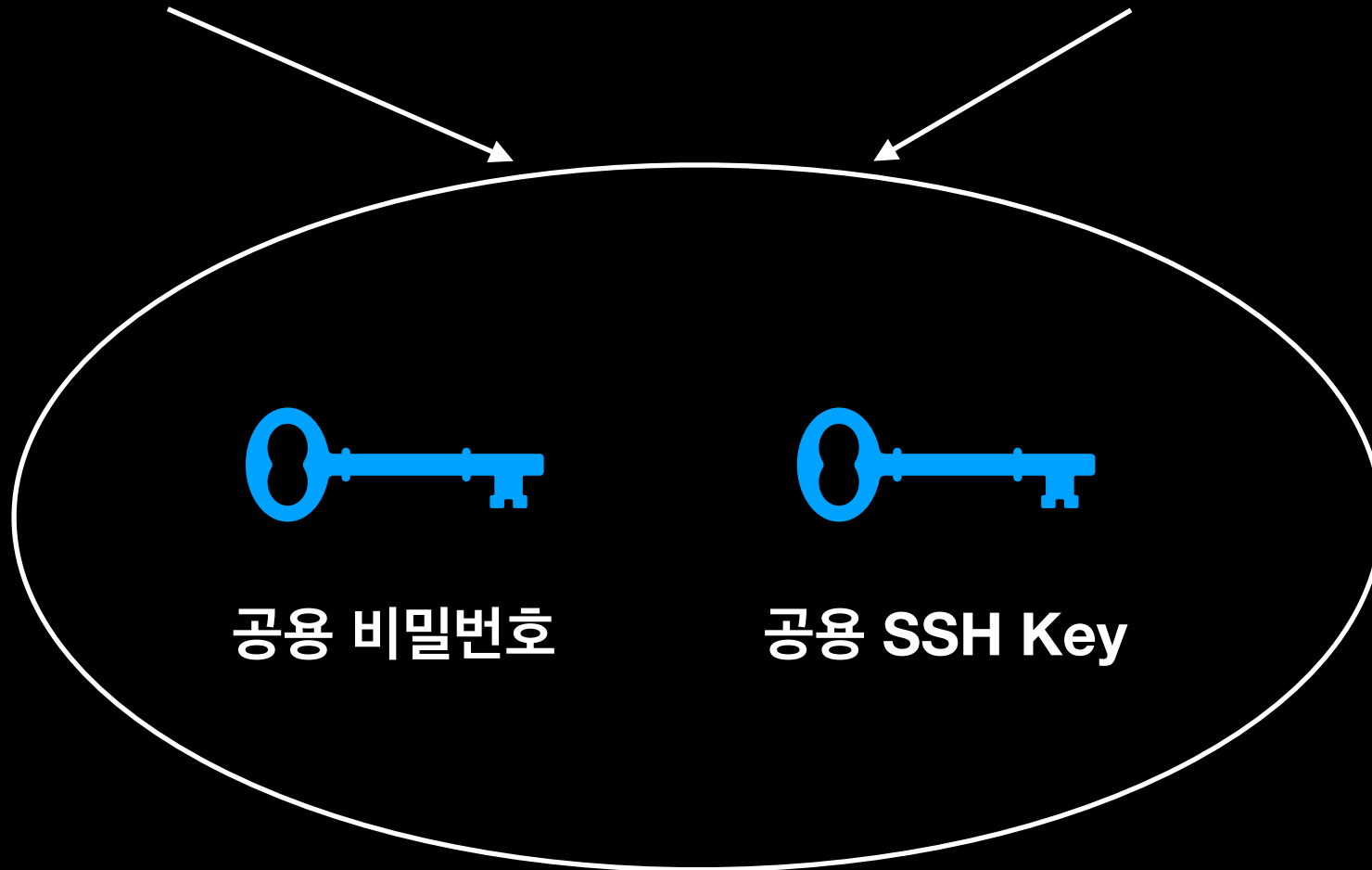
DEVSISTERS 김도윤, 김민규

CookieRun
OvenBreak

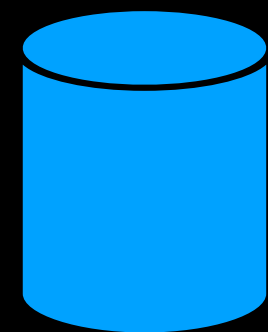
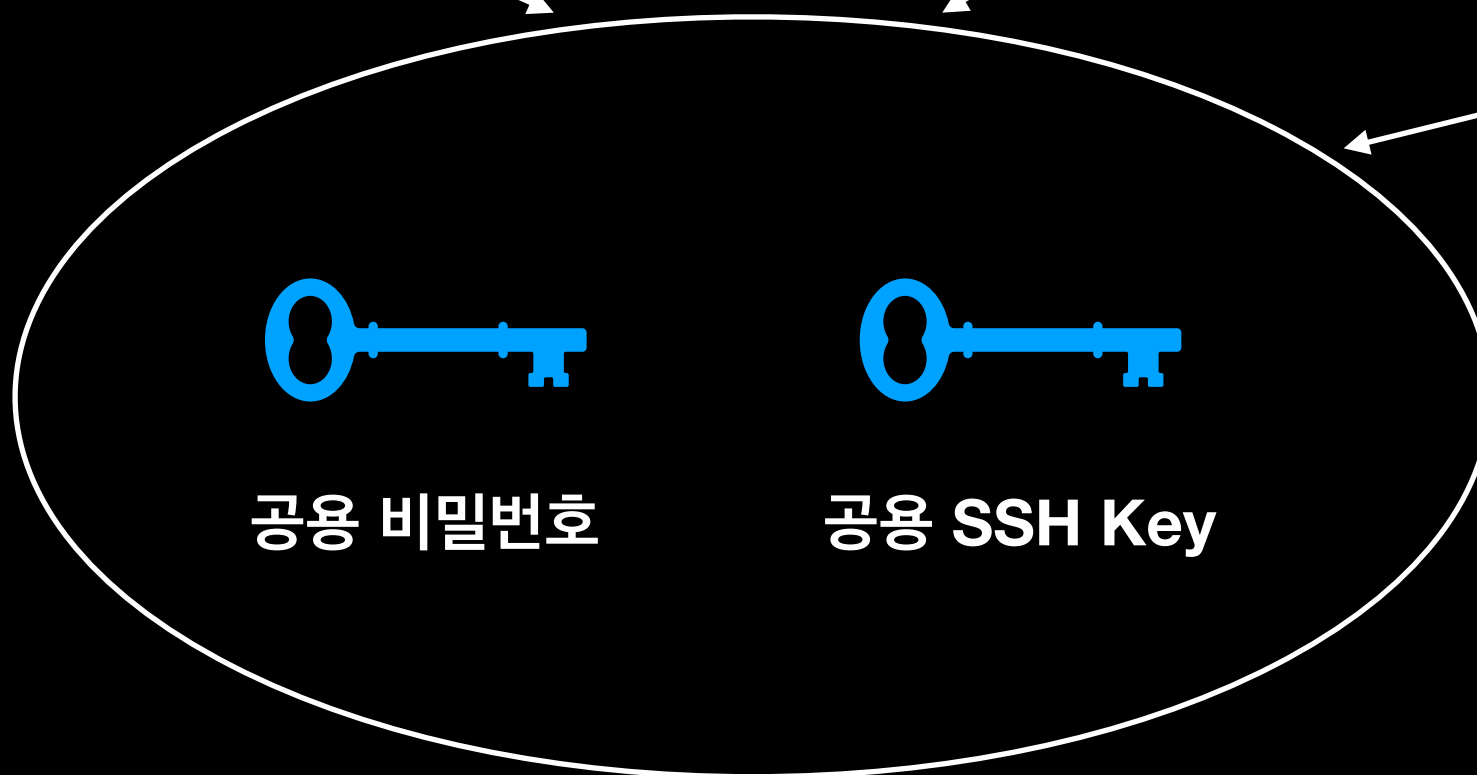
HashiCorp

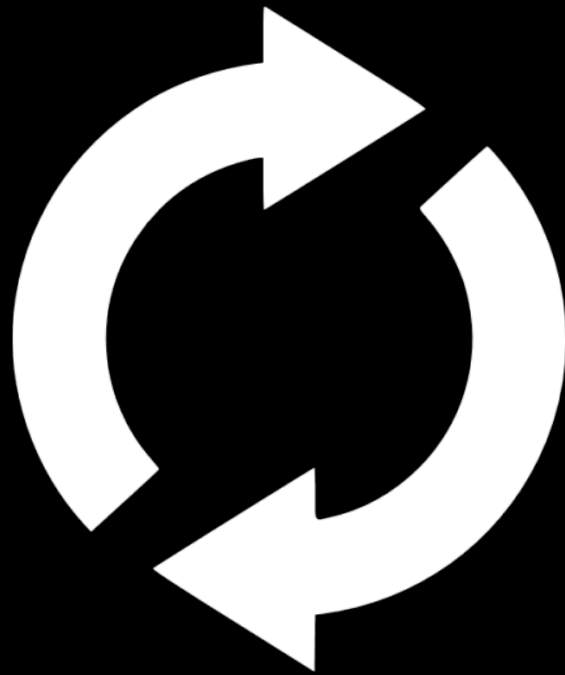
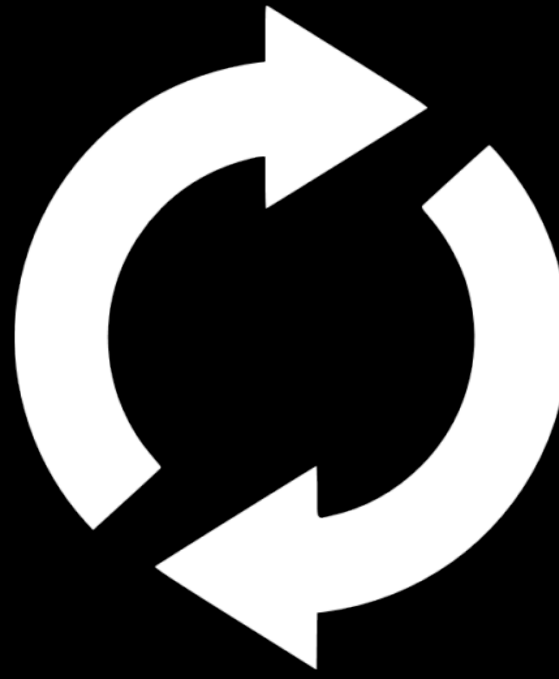
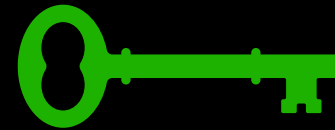
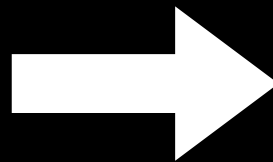
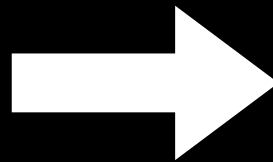
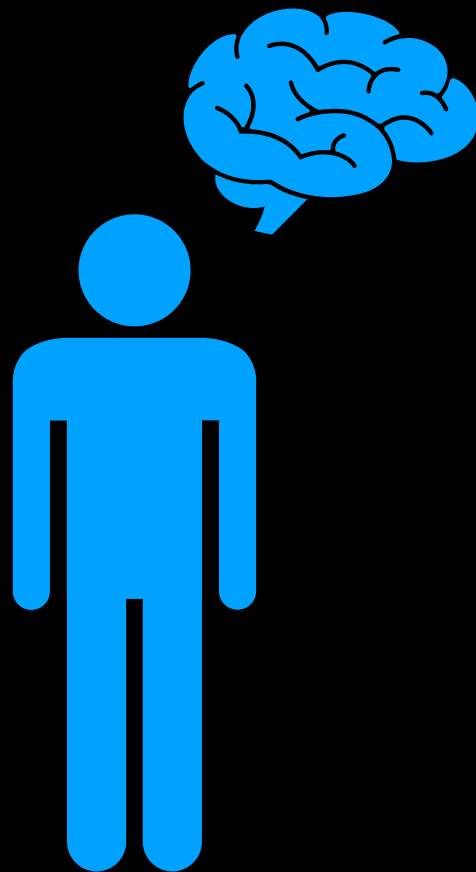
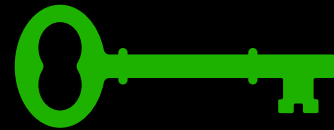
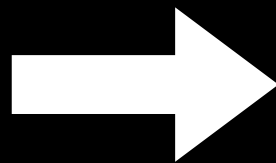
Terraform

HashiCorp
Vault

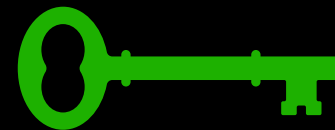
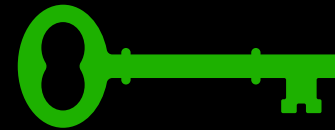
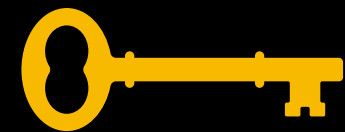
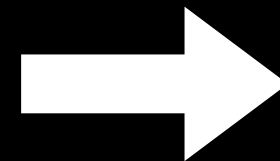
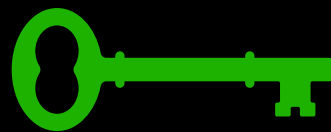
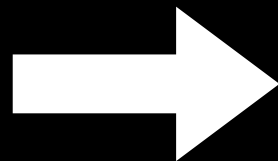
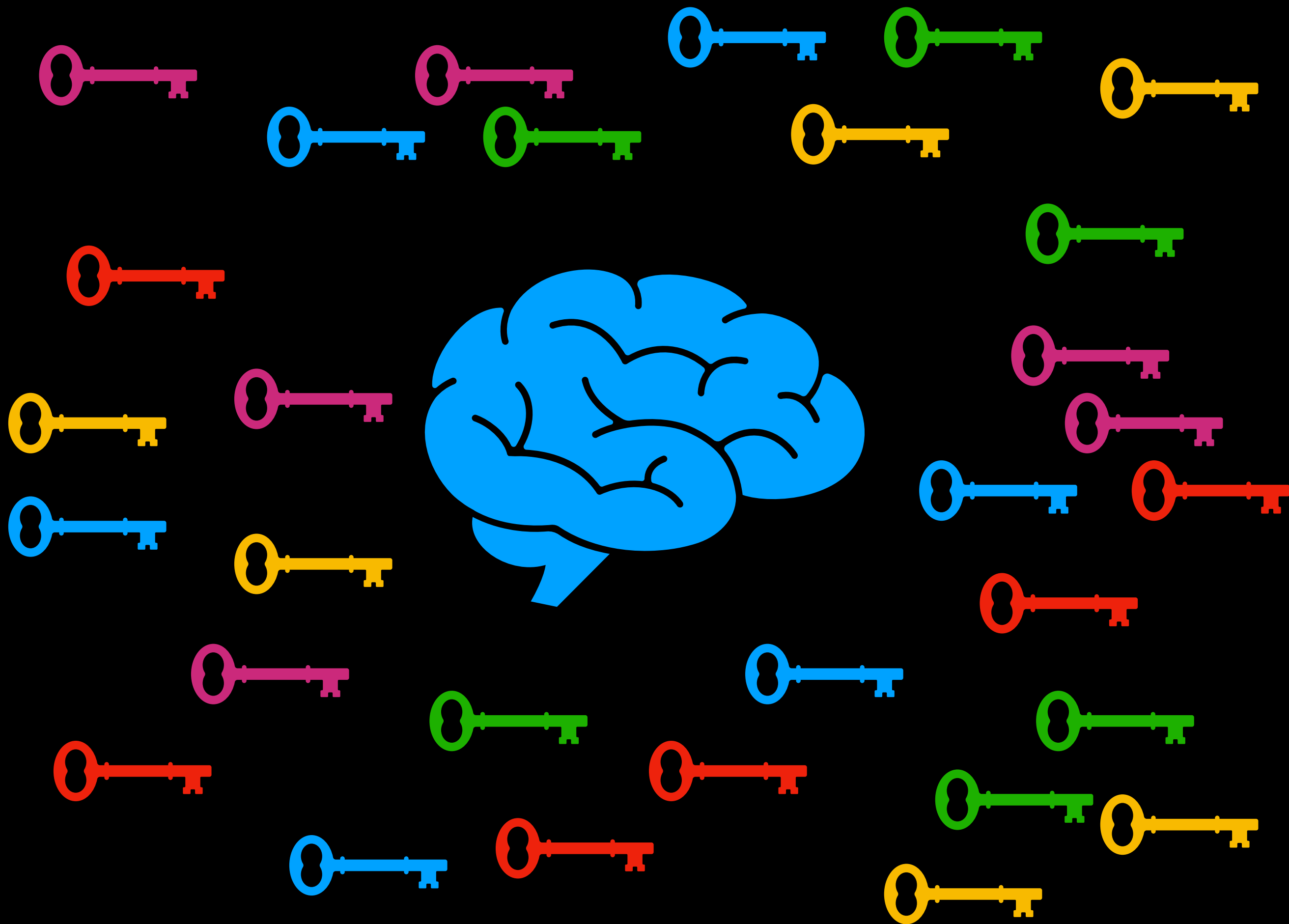
< 10명 시절

입사자 선물세트

공용 SSH Key

서버 100대
사람 10명
공용 비밀번호
공용 SSH Key

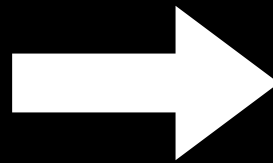
어느날...

퇴사자

서버 100대
사람 9명
퇴사자
공용 비밀번호
공용 SSH Key

Key Rotation

super_secret_password → hyper_safe_passw0rd

머릿속 비밀번호 교체

어떻게 전달하죠?

Slack
super_secret_passwo
hyper_safe_passw0rd

귓속말

우리 비밀번호는 이제 B야 (소곤소곤)
(실화)

super_secret_password
hyper_safe_passw0rd
ultra_secret_passw0rd

SSH Key 교체

100번만 하면 됩니다.

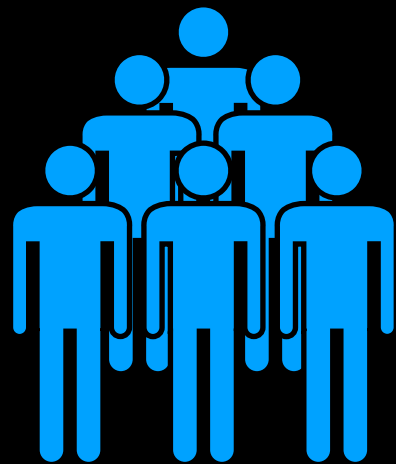
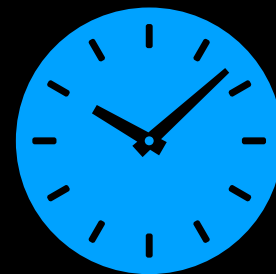
old_ssh_key

new_ssh_key

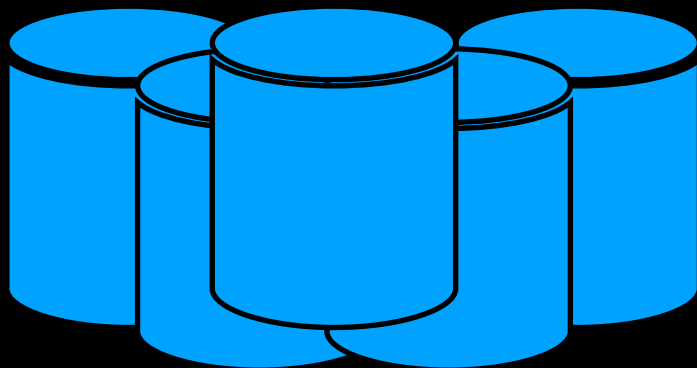
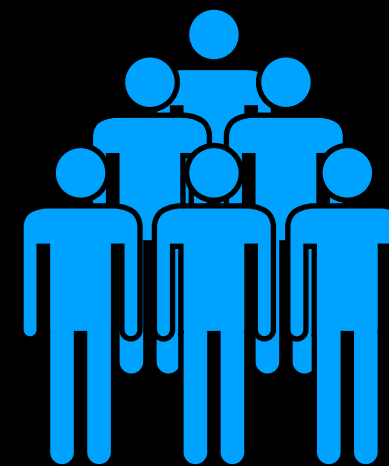
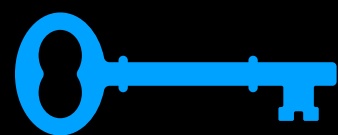
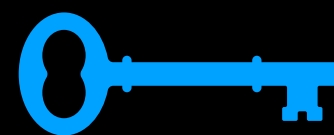
Q: 스크립트 못만들어요?

AWS 계정이 달라요

OS가 달라요 (Ubuntu, Debian, Amazon Linux)

잘못 돌리면 엄청골치아파요 (SSH 안됨)

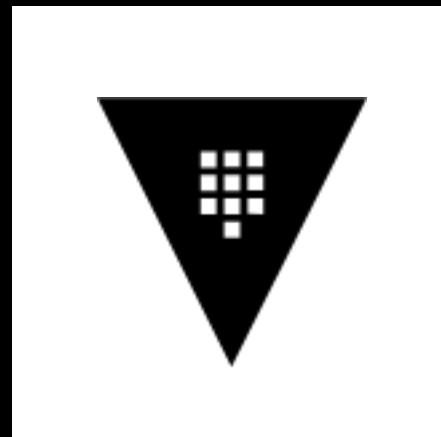
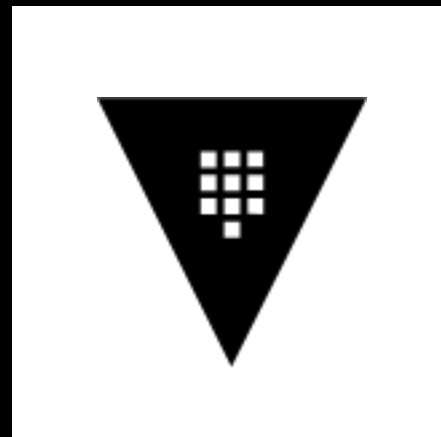
키 교체 TF 결성

6-7명

약 3-4주

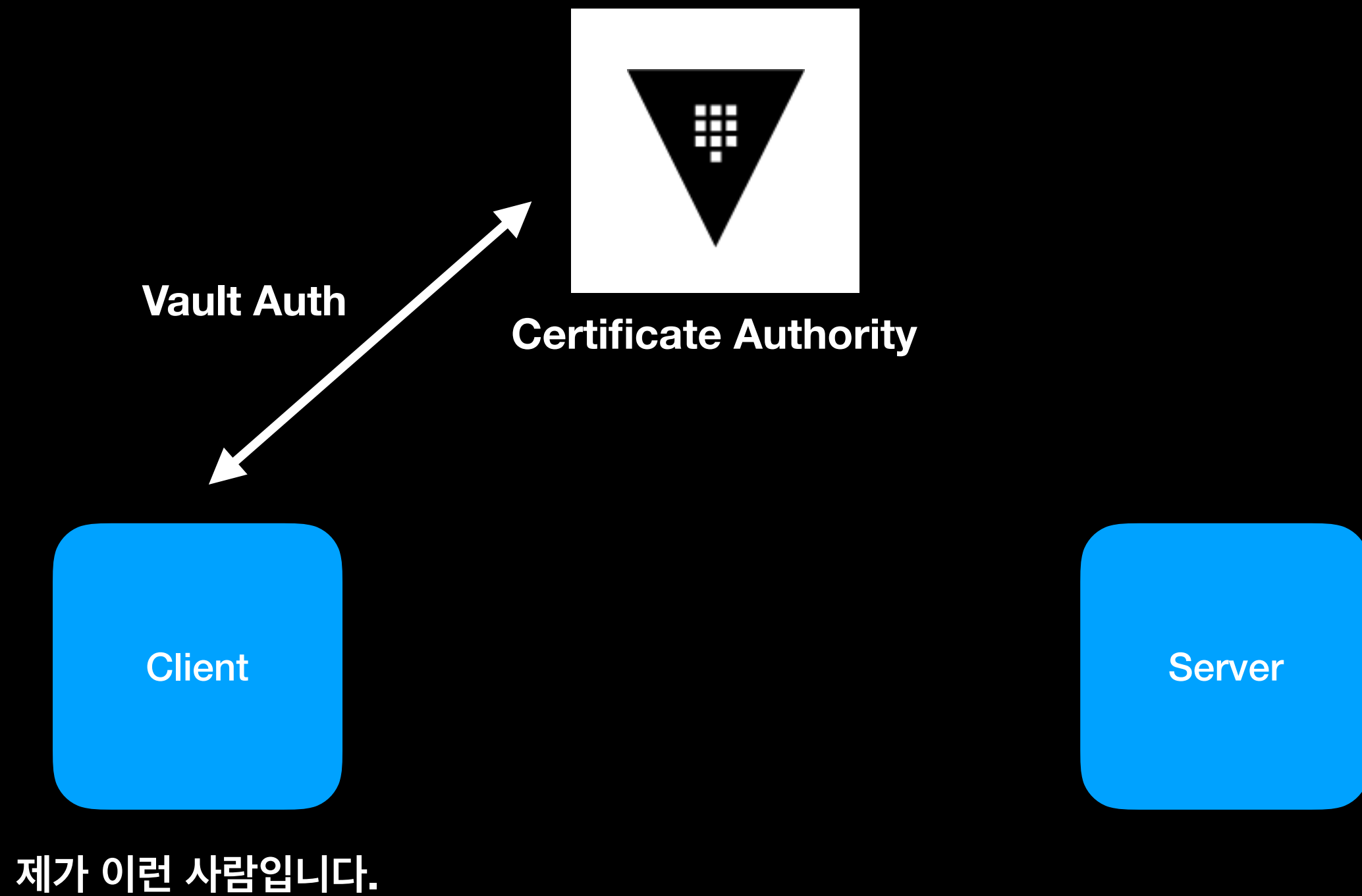
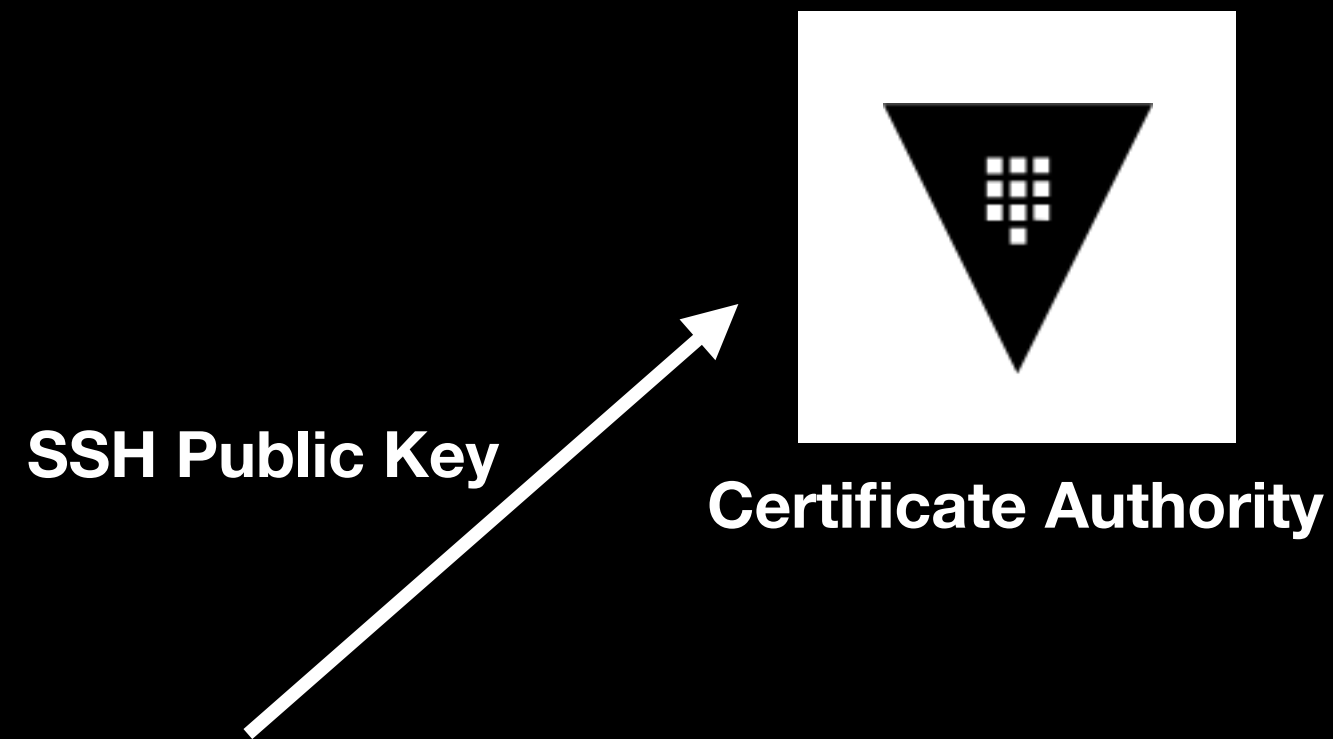
20명+

서버 500+대
사람 20+명
공용 비밀번호
공용 SSH Key
AWS
AWS 계정 5+개
AWS VPC 10+개

찰스홍
난 죽음을
택하겠다!
상대 턴

HashiCorp

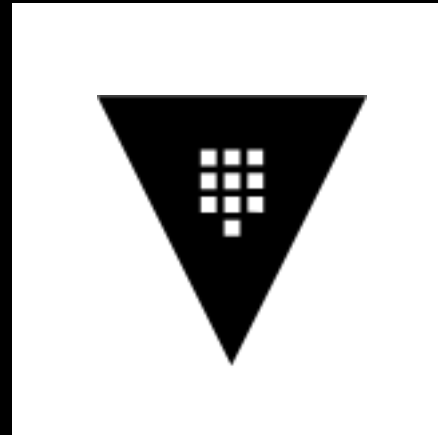
Vault

Vault를 통한 SSH

SSH Certificate Authentication

(믿을 수 있는) Secret Store

(훌륭한) Certificate Authority

Vault Auth
Certificate Authority
Client
Server
제가 이런 사람입니다.

SSH Public Key
Certificate Authority
Client
Server
제가 이런 사람인거 아시죠?
접근을 허가해 주세요!

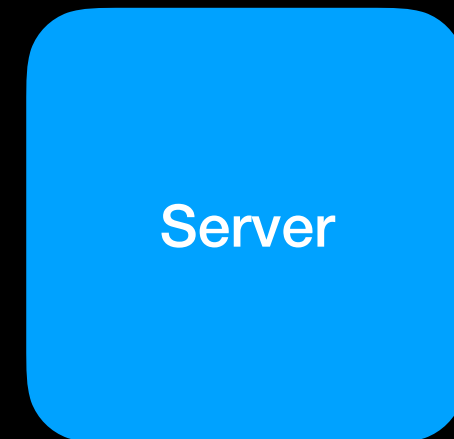
접근을 허가한다.
5분 안에 접속하시오.
Signed
SSH Public Key
Certificate Authority
Client
Server

Certificate Authority

제가 접속해도 된대요.

Certificate Authority

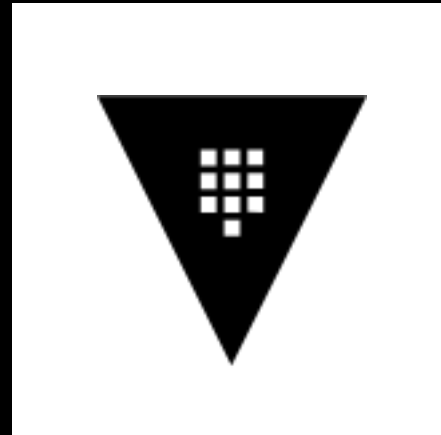
ca.pub

cf) 보통은 public key를 전달받아
authorized_keys와 대조한다.

Client

Server

서명된 것이 맞네요.
계속 진행해도 좋습니다.

Certificate Authority

Signed
SSH Public Key
퇴사자
Server
제가 접속해도 된대요^^

Certificate Authority

퇴사자

T_T

Server

아이고 서명은 됐는데
시간이 지나셨네요.

초간단 서버 셋업

- 인증서를 만든다.

```
$ vault write ssh-client-signer/config/ca generate_signing_key=true
Key             Value
---             -----
public_key      ssh-rsa AAAAB3NzaC1yc2EA...
```

초간단 서버 셋업

- 역할을 등록한다.

```
$ vault write ssh-client-signer/roles/my-role -<<"EOH"
{
  "allow_user_certificates": true,
  "allowed_users": "*",
  "default_extensions": [
    {
      "permit-pty": ""
    }
  ],
  "key_type": "ca",
  "default_user": "ubuntu",
  "ttl": "30m0s"
}
EOH
```

초간단 서버 셋업

- 서버에 인증서 공개키를 등록한다.

```
# /etc/ssh/sshd_config
# ...
TrustedUserCAKeys /etc/ssh/trusted-user-ca-keys.pem
```

초간단 서버 셋업

- 공용 AMI로 굽는다.

- 끝!

더 간단한 클라이언트 접속

```
$ vault ssh -mode=ca -role=my-role user@1.2.3.4
```

간단하지만 강력한 관리

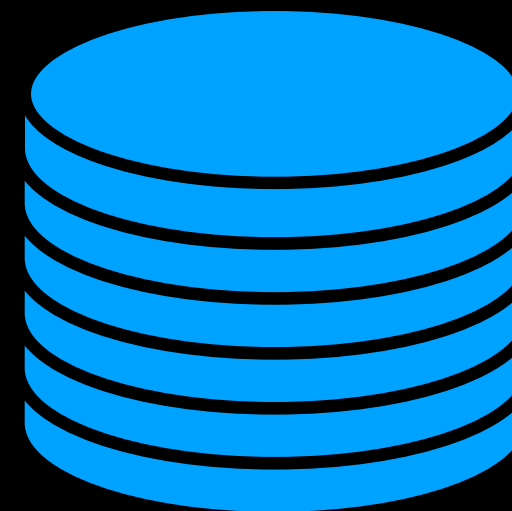
- Vault 자체의 훌륭한 권한 관리 시스템을 활용
- 개인이나 그룹에 따라 접근 가능한 인스턴스 설정 가능
- 새 팀원이 오면 필요한 그룹에 넣어주기만 하면 된다.

FAQ

- Vault에 장애가 나면?
  - 비상용 키
  - 서버들에 기본으로 public key를 등록
  - 암호화된 private key는 안전한 금고 등지에 보관한다.

자세한 내용은

- HashiCorp 제품의 장점: 좋은 문서화
- https://www.vaultproject.io/docs/secrets/ssh/signed-ssh-certificates.html
- ~~옛날엔 이런 거 없었는데..~~

Application에
Secret 넘기기

db_password
data
Application
Database

어떻게 알려주죠?

Password in Code

```
DB.connect(user: 'roo                    uper_secure_password')
```

imfishhead/missionz – .env

Showing the top match   Last ind

```
1  AWS_BUCKET=missionz
2  AWS_ACCESS_KEY=AK
3  AWS_SECRET_KEY=
```

구체적으로는요

- Code의 접근 권한과 Secret의 접근 권한이 다른 경우가 많음.
- Rotate하면 코드 싹다 갈아엎어야
- 일단 Github도 회사 밖이다! (엔터프라이즈는 비싸요)

Workarounds

AMI에 패스워드 ... 때는 직접 입력

서버에만 올려두 ... 커밋하지 않음

작업자가 모르는게 최고

Vault KV

```
db_pw = Vault.logical.read("/secret/db_password")
DB.connect(user: 'root', password: db_pw)
```

```
DB_USER=$(vault read -format=json "secret/app/db_credential" | jq -r '.data.username')
DB_PW=$(vault read -format=json "secret/app/db_credential" | jq -r '.data.password')
```

Q. Vault는 뭘 믿고 얘한테 키를 줘요?

고종황제가 사용했던 국새(國璽)

AWS에서 찍어주는 옥새

인스턴스 자격 증명 문서

인스턴스 자격 증명 문서는 인스턴스를 설명하는 JSON 파일입니다. 인스턴스 자격 증명 문서에는 문서에 제공된 정보의 정확도, 오리진 및 신뢰성을 확인하는 데 사용할 수 있는 서명 및 PKCS7 서명이 함께 제공됩니다. 예를 들어, 유료 업데이트가 포함된 무료 소프트웨어를 다운로드했을 수 있습니다.

인스턴스 자격 증명 문서는 인스턴스를 시작할 때 생성되고 인스턴스 메타데이터를 통해 인스턴스에 공개됩니다. 이 문서는 인스턴스의 속성(예: 구독 소프트웨어, 인스턴스 크기, 인스턴스 유형, 운영 체제, AMI 등)이 유효한지 검사합니다.

> **중요**
>
> 인스턴스 자격 증명 문서와 서명은 동적인 특성을 지니고 있기 때문에 규칙적으로 인스턴스 자격 증명 문서와 서명을 검색하는 것이 좋습니다.

https://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/instance-identity-documents.html

Amazon Cloud

EC2 Metadata Service

Public EC2 API

1

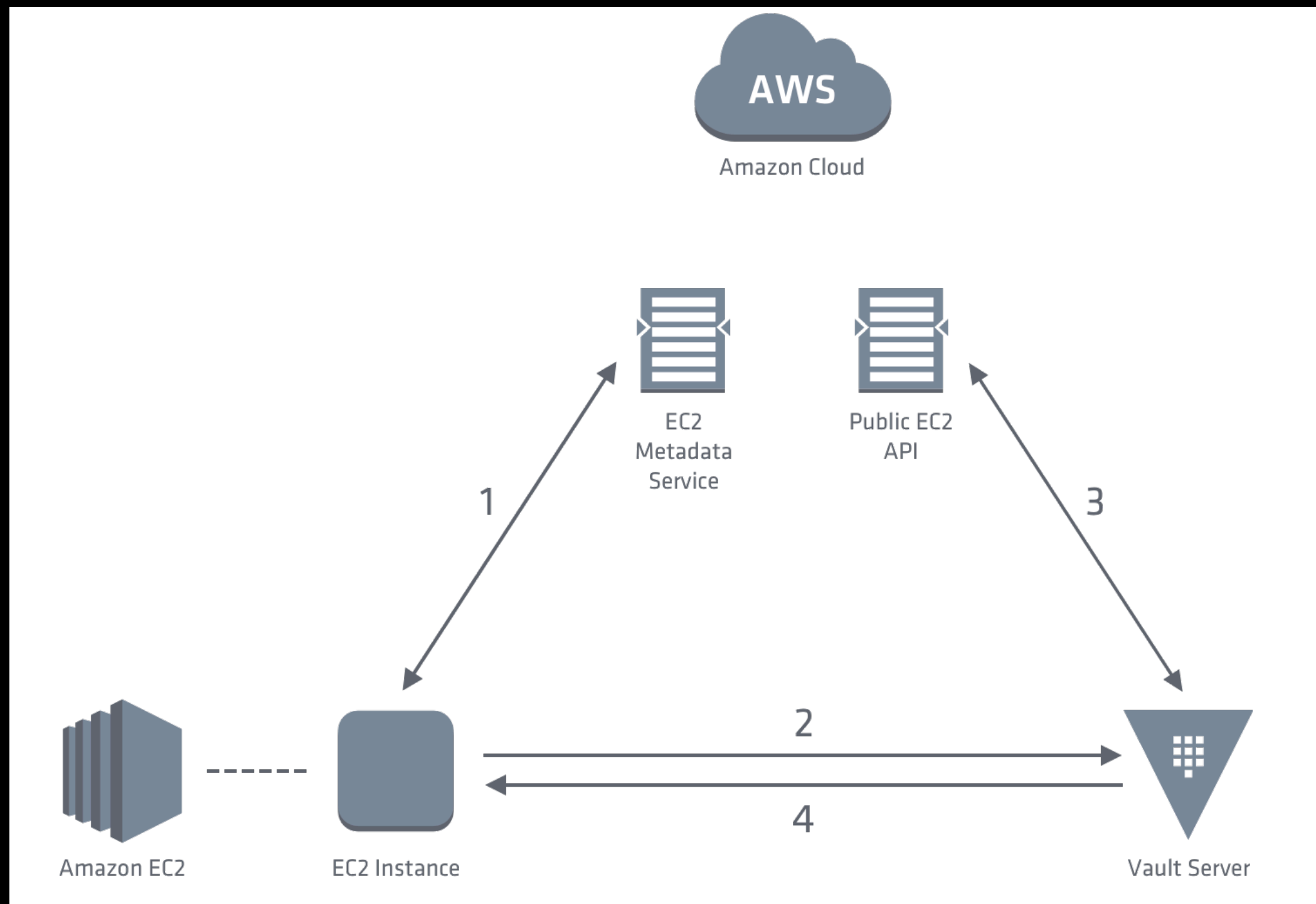
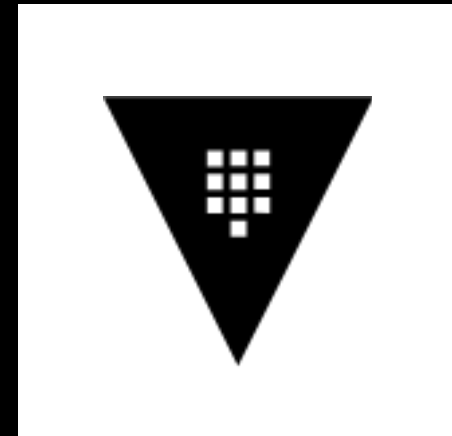
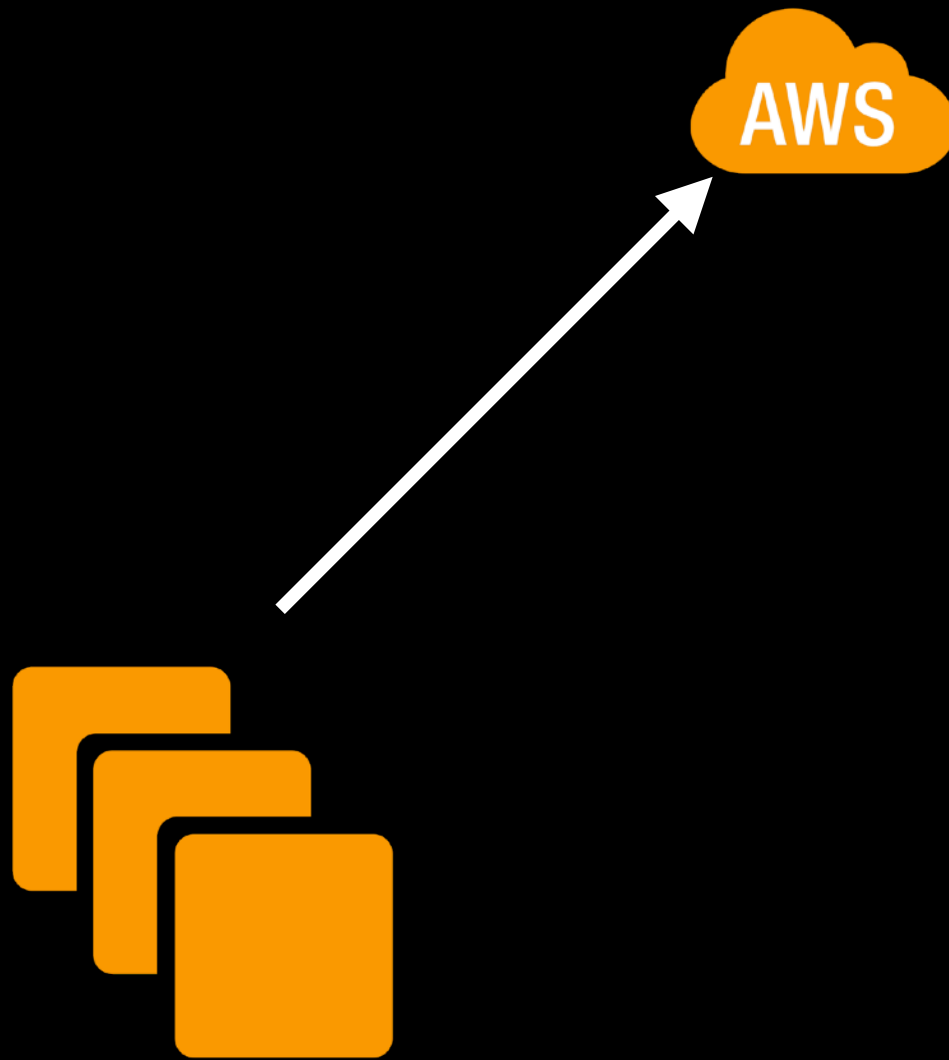
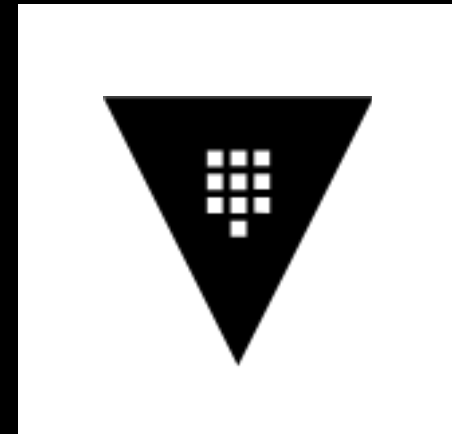
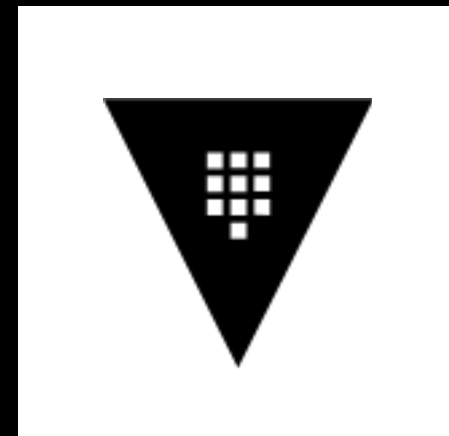
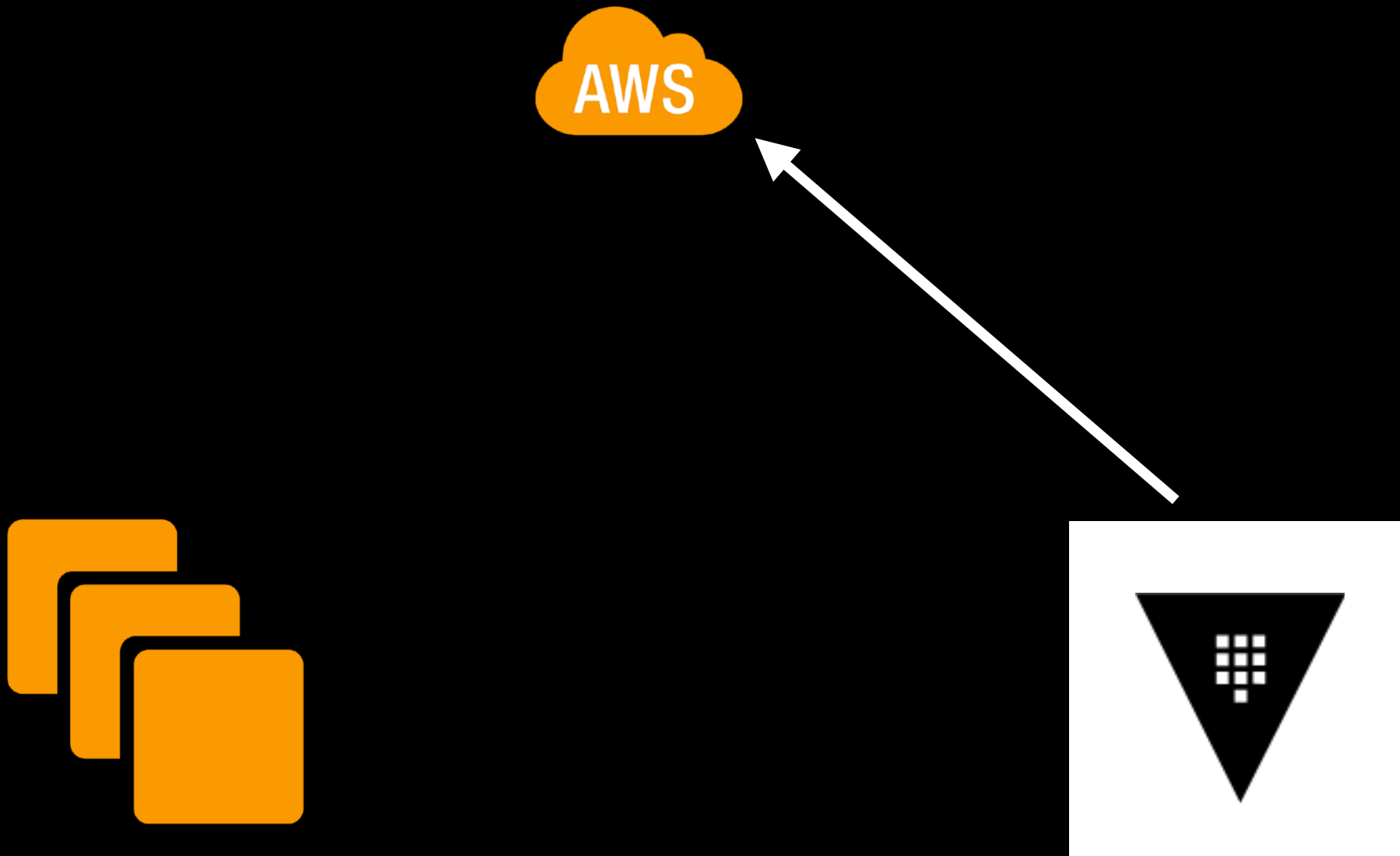
3

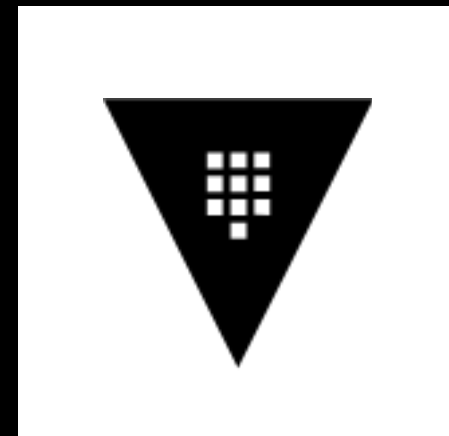
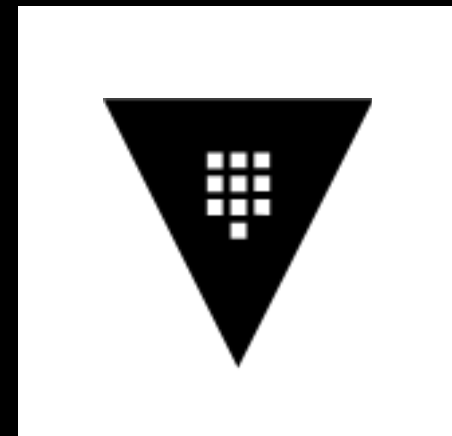
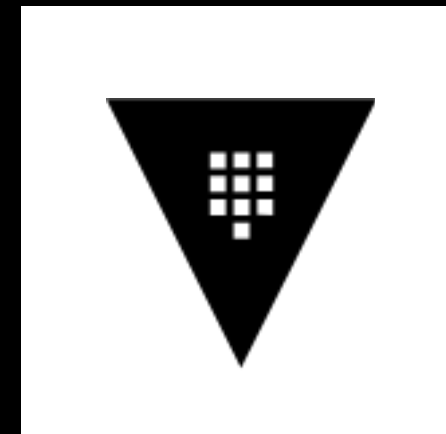
2

4

Amazon EC2

EC2 Instance

Vault Server

신분증명서 떼어주세요

옛다
AWS
자격 증명 문서

Vault 열람 신청합니다

이거 당신이 서명한거 맞아요?

ㅇㅇ 제 서명 맞아요
추가로 얘 Tag / IAM Role/ Metadata도 알려드릴게요.

당신 메타데이터가 이러저러하니
열람할 수 있는 토큰을 드리죠

당신 메타데이터가 이러저러하니
열람할 수 있는 토큰을 드리죠

vault_auth.sh

```
PKCS7="$(curl http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 | tr -d '\\n')"
vault write -address="$VAULT_ADDR" -field=token auth/aws-ec2/login role="$VAULT_ROLE" pkcs7="$PKCS7" nonce="$(cat /proc/sys/kernel/random/uuid)"
```

- 역시 Packer로 모든 인스턴스에 구워넣음
- Vault Agent (3일 전에 나옴)

FAQ

- PKCS를 가로채이면 어쩌죠?
  - Trust on first use
  - 두번째 login부터는 첫번째에 제공한 nonce값 필요

FAQ

- 이제는 패스워드를 100% 완벽하고 안전하게 관리하고 있나요?
  - 100%는 아니다
  - PASSWORD를 읽어서 docker ENV로 넘기는데,
    썩 안전한 장소는 아님.
  - 제대로 하려면 각 어플리케이션이 Vault API를 호출해야 함

Vault 구축 과정

아래 상황은 픽션입니다

- 아 이거 도입했는데 장애도 너무 많이 나고 일할때 불편해요
- 우리 팀은 저거 안 쓸래요 뭐가 좋은지 잘 모르겠어요
- 이거 너무 셋업 어려워요 어떻게 쓰는지 모르겠어요
- 이런 복잡한거 말고 하던대로 하면 안돼요?

구축시 신경쓸 점

- 안정성
  - Vault의 갑작스런 장애를 인간도 기계도 원하지 않는다.
  - 업데이트가 잦다. 무중단 업데이트 가능할 필요성

구축시 신경쓸 점

- 보안
  - 절대 털리면 안될 곳이므로 최대한 안전하게
  - Vault 권장사항을 정독하고 충실히 따름

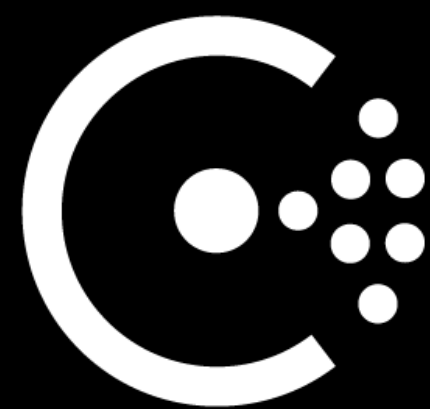
구축시 신경쓸 점

- 사용성
  - 사람들의 생활 습관을 최대한 적게 바꾸는 방향으로
  - 사람들이 적응하는데 어렵지 않도록

안정성

데이터 저장

일단 지원되는 건 엄청 많음

- https://www.vaultproject.io/docs/configuration/storage/index.html
  - MySQL
  - DynamoDB
  - CockroachDB
  - Filesystem
  - ....

HashiCorp
Consul

Consul

- KV Store, Service Discovery, etc..
- 이미 사내 사용 및 구축 사례 있음

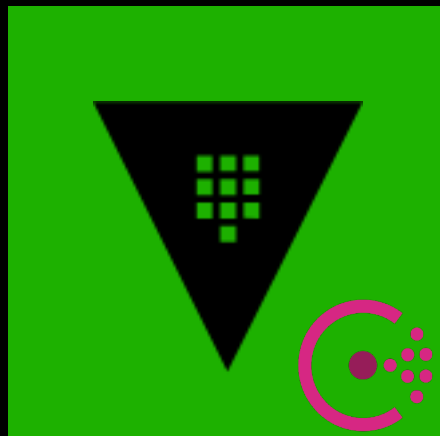
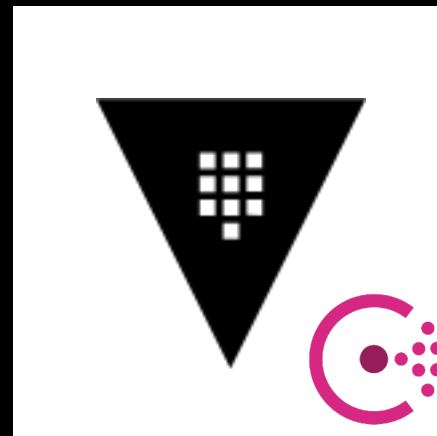
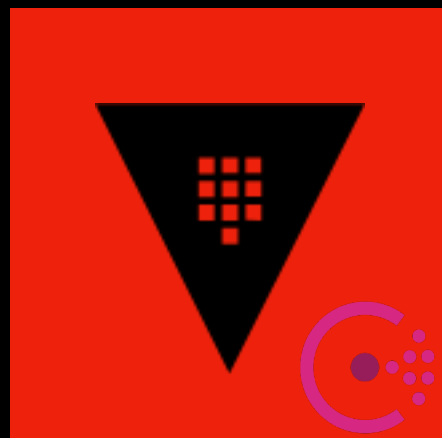
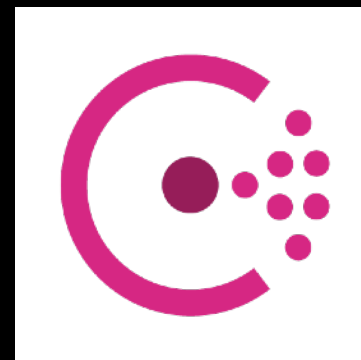
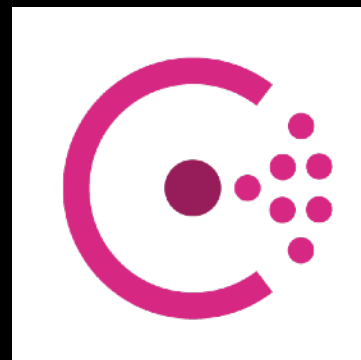
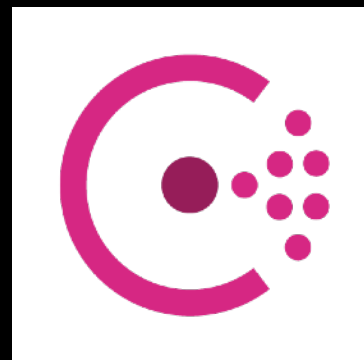
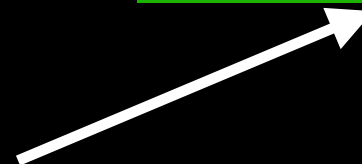
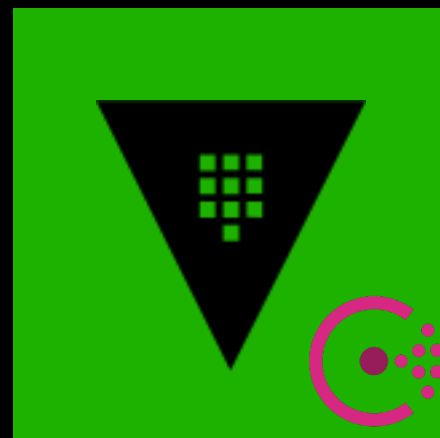
+ 뭔가 믿음직함 (중요)

- **High Availability** – the Consul storage backend supports high availability.
- **HashiCorp Supported** – the Consul storage backend is officially supported by HashiCorp.

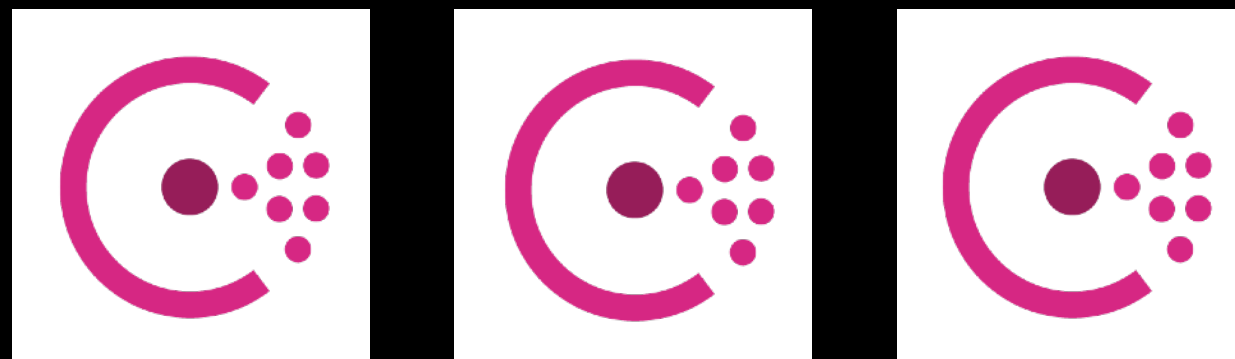
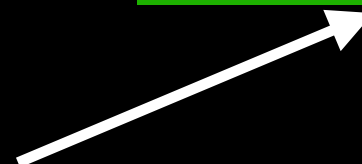
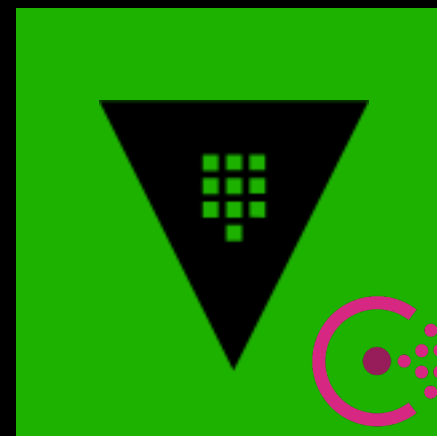
고가용성

서버가 두 대

Active-Standby

Active-Standby

Active-Standby

ELB

Failed

Active

Consul Cluster

Consul

- 이미 훌륭한 Service Discovery 기능 제공
  - 유저가 DNS 설정을 하던가
  - 외부 서비스를 활용 (fabio, etc..)

ELB

- 사실 세팅은 Health Check 정도면 됨
- 관리 코스트가 몹시 적음
- 유저에게도 뭔가 시키지 않는 선택지
- 아쉽지만 사용성이 좋았다.

까다로울 것 같다면

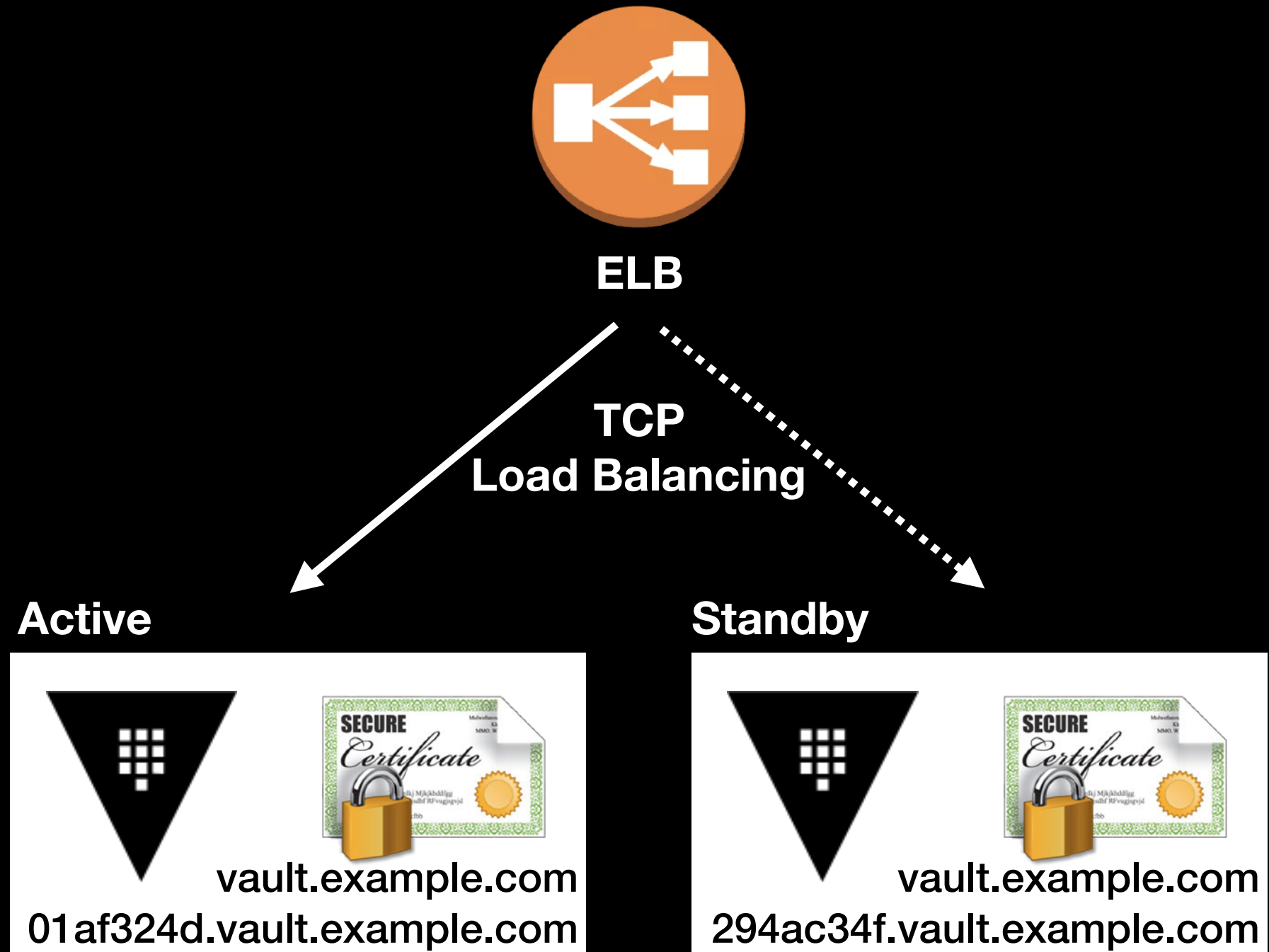
- 정말 친절한 레퍼런스 가이드 문서
- https://www.vaultproject.io/guides/operations/vault-ha-consul.html
- ~~이것도 옛날엔 없었는데…~~

보안

할거면 제대로 해야지 뭐

End-to-End TLS

Active

Standby

인증서

- 각각의 인스턴스에서 알아서 발급, 알아서 갱신
- Let's Encrypt!
- https://github.com/Neilpang/acme.sh
- Route53을 활용한 DNS-01 인증

Audit Logging

- 현재는 logrotate를 이용해 주기적으로 s3에 적재
- syslog도 사용 가능
- 로깅 인프라가 있다면 (ELK 등) 충분히 활용 가능

사용성

문서화

정책, 정책, 더 많은 정책

- 셋업은 한때일 뿐 권한 설정에 더 많은 시간을 쓰게 된다.
- 이미 있는 인프라를 활용하면 절약 가능
    - ex) GitHub 그룹, LDAP

Secret 관리

- 체계적이고 직관적인 경로 설정
  - 예) /secret/app/appname/rds
- 정책에 관한 꼼꼼한 문서화
- 가급적 도입 전부터!

적응에 도움주기

- 개념 자체가 생소하고 어려운 물건이다.
- 기본 개념부터 하나하나 문서화
- 회사를 한 바퀴 돌아다니면서 강의 세션 진행
- 각종 도움되는 스크립트 작성 및 공유

회고

정말 편하다!!

그런거 말고

예상을 벗어났던 것

권장은 권장일 뿐

- End-to-End TLS같은걸 하게 되면 셋업 난이도가 **수직상승**
- 내줄 건 과감하게 내주자. 다음에 해도 된다.

Consul

- 은근 관리가 까다롭다
- 다수의 업데이트 실패 사례 있음
- 사용한다면 적극적인 백업을 권장

히어로

- 없던 정책을 새로 설정하고 관리하는 것은 생각보다 힘든 일
- 결국 의지와 추진력을 가진 자가 필요하다.

Case: Unseal

- Vault 인스턴스는 처음 뜨면 봉인된 상태
- 지정된 n명 중 m명이 모이면 해제할 수 있다.
- 7인의 Unsealers를 지정해서 관리 책임을 분산..!!

이상

아이언 맨
로버트 다우니 주니어
토르
크리스 헴스워스
헐크
마크 러팔로
캡틴 아메리카
크리스 에반스
블랙 위도우
스칼렛 요한슨
호크 아이
제레미 레너
로키
톰 히들스턴
닉 퓨리
사무엘 L. 잭슨
최강의 슈퍼히어로들이 모였다
MARVEL
어벤져스
AVENGERS.CO.KR
4월 26일 대개봉
IN 3D, real D 3D AND IMAX 3D
MARVEL

현실

혼자서는 팀을 구할 수 없다

- 슈퍼맨 하나로는 역부족
- 애매하게 분산된 책임
- Unseal 작업은 둘 이상이 필요
- 결국 운영하는 두 명이 Unseal도 하는 중

인내심

- Vault는 굉장히 활발히 개발되고 있는 프로덕트
- 목빠지게 기다리던 업데이트도 문제가 생겨 종종 롤백
- **새로 개발되는 기능은 좀 기다렸다 써보자.**

또는 행동력

- 아니면 PR과 이슈를 작성할 각오를 해야..

예상보다 더 좋았던 것

Audit Logging

- Vault 최고의 장점이자 핵심 가치
- 조사하면 정말 뭐든지 다 나온다

Audit Logging

- 이 비밀번호 누가 이렇게 바꿔놨어요? -> 검거
- 그렇게 한 적이 없는 것 같은데 세팅이 이렇네요 -> 검거

일단 넣어

- 컨피그 파일이 있는데.. -> KV에 일단 넣어
- 공용 OTP는.. -> TOTP Backend
- 그냥 간단히 안전하게 뭘 전달하고 싶은데.. -> Transit Backend

일단 넣어

- 심지어 이게 극히 일부만 쓰고 있는 것

엄청난 발전 속도

- 각종 사용자 가이드 문서
- Web UI
- PROXYv1 / X-Forwarded-For support
- Secret Versioning
- CLI 자동완성
- 기타등등...

엄청난 발전 속도

- 1년도 안 되어 어마어마한 발전을 이룬 제품
- 체인지로그를 기다리는 맛이 있다.
- 이젠 충분히 쓸만하다!

줄줄이 패키지

- 자사 제품과의 뛰어난 연계
- Consul
- Consul Template
- Terraform
- Packer
- Nomad

이런 분들에게 추천합니다

- Terraform을 적극적으로 사용하는 팀
- 보안 스탠다드가 높거나 높이고 싶은 팀
- 급격히 성장중이고 빠르게 인원이 증가하는 팀

DEVSISTERS

devsisters.com/jobs

끝